Assignment 1:

1.How internet works?

The Internet is a vast global network that connects millions of computers, people, and other devices from around the world. It allows us to access information from anywhere in the world, send messages instantly, and interact with each other online.

The Internet works by connecting networks together through a series of routers and switches. A router forwards packets of data between different networks while a switch links devices within a single network. This enables computers to communicate with each other and access content stored on remote servers.

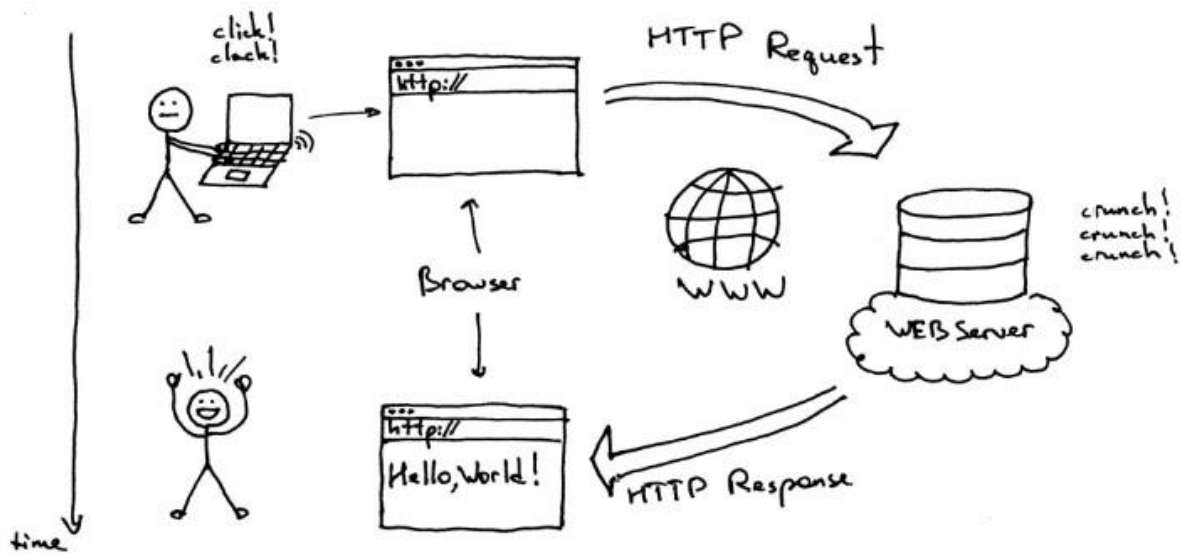
In networking, a packet is a small segment of a larger message. Each packet contains both data and information about that data. The information about the packet's contents is known as the "header," and it goes at the front of the packet so that the receiving machine knows what to do with the packet. To understand the purpose of a packet header, think of how some consumer products come with assembly instructions.

When data gets sent over the Internet, it is first broken up into smaller packets, which are then translated into bits. The packets get routed to their destination by various networking devices such as routers and switches. When the packets arrive at their destination, the receiving device reassembles the packets in order and can then use or display the data.

Because all Internet-connected computers and other devices can interpret and understand these protocols, the Internet works no matter who or what connects to it.

2. How browser works?

Browsers are responsible for retrieving and displaying web content to users. When a user enters a URL or clicks on a link, the browser initiates a complex series of actions to retrieve the web content from a server and display it on the user's device.



The process begins with Domain Name System (DNS) resolution, where the browser translates the domain name into an IP address to locate the server where the web page is stored.

- The browser then sends an HTTP request to the server, specifying the path and parameters of the requested resource.
- Once the server receives the request, it sends an HTTP response to the browser containing the requested resource in HTML, CSS, and JavaScript code.
- The browser's rendering engine interprets and renders the code to display the web page on the user's device.
- The CSS stylesheets are applied to format the web page's content, including fonts, colors, and layout.
- The browser may also execute JavaScript code on the web page to add interactivity and dynamic behavior.

As new content is loaded or changes are made to the web page, the browser updates the display accordingly.
Apart from the working principles, there are a few terms related to browsers that you must know.

3. What is Server?

A server is a hardware device or software that processes requests sent over a network and replies to them. A client is the device that submits a request and waits for a response from the server. The computer system that accepts requests for online files and transmits those files to the client is referred to as a “server” in the context of the Internet.

A Server is a program or a device that provides functionality for called clients which are other programs or devices. This architecture is called the client-server model.

A single overall computation is distributed across multiple processes or devices. Servers can provide various functionalities called services. These services include sharing data or resources among multiple clients or performing computations for a client. Multiple clients can be served by a single server, and a single client can use multiple servers.

4. what are the types of server available?

1. Application Server

These servers host web apps (computer programs that run inside a web browser) allowing users in the network to run and use them preventing the installation of a copy on their own computers. These servers need not be part of the World Wide Web. Their clients are computers with a web browser.

2. Catalog Server

These servers maintain an index or table of contents of information that can be found across a large distributed network. Distributed networks may include computers, users,

files shared on file servers, and web apps. Examples of catalog servers are directory servers and name servers. Their clients are any computer program that needs to find something on the network. An example can be a domain member attempting to log in, an email client looking for an email address, or a user looking for a file

3. Communication Server

These servers maintain an environment needed for one communication endpoint to find other endpoints and then communicate with them. These servers may or may not include a directory of communication endpoints and a presence detection service, depending on the openness and security parameters of the network. Their clients are communication endpoints.

4. Computing Server

These servers share vast amounts of computing resources which include CPU and random-access memory over a network. Any computer program that needs more CPU power and RAM than a personal computer can probably afford can use these types of servers. The client must be a networked computer to implement the client–server model which is a necessity.

5. Database Server

These servers maintain and share any form of database over a network. A database is an organized collection of data with predefined properties that may be displayed in a table. Clients of these servers are spreadsheets, accounting software, asset management software, or virtually any computer program that consumes well-organized data, especially in large volumes.

6. Fax Server

These servers share one or more fax machines over a network which eliminates the hassle of physical access. Any fax sender or recipient is the client of these servers.

7. File Server

Shares files and folders, storage space to hold files and folders, or both, over a network. Networked computers are the intended clients, even though local programs can be clients.

8. Game Server

These servers enable several computers or gaming devices to play multiplayer games. Personal computers or gaming consoles are their clients.

9. Mail Server

These servers make email communication possible in the same way as a post office makes snail mail communication possible. Clients of these servers are senders and recipients of email.

10. Print Server

These servers share one or more printers over a network which eliminates the hassle of physical access. Their clients are computers in need of printing something.

11. Proxy Server

This server acts as an intermediary between a client and a server accepting incoming traffic from the client and sending it to the server. Reasons to use a proxy server include content control and filtering, improving traffic performance, preventing unauthorized network access, simply routing the traffic over a large and complex network. Their clients are any networked computer.

12. Web Server

These servers host web pages. A web server is responsible for making the World Wide Web possible. Each website has one or more web servers. Their clients are computers with a web browser.

5. What is SEO? Importance of SEO?

SEO is the practice of increasing the quantity and quality of traffic to your website through organic search engine results. A higher rank when someone searches a term in your industry increases your brand's visibility online. The increase in visibility will drive more organic traffic to your site, and this, in turn, gives you more opportunities to convert qualified prospects into customers. When done

correctly, SEO can help your brand stand above others as a trustworthy company and further improve the user's experience with your brand and website.

SEO is important for brands as it's a highly effective way to improve your brand's visibility through search, drive more traffic to your website, establish your brand as a trusted authority in your industry, sustainably and reliably grow your business, and much more. Here's how each of these factors contributes to the importance of SEO for your brand.

Visibility and Rankings

When searching for a service or product online, users are more likely to choose one of the top five suggestions that the search engine shows them. After all, while Google may return thousands upon thousands of search results for any given term, the vast majority of searchers never make it past the first page, and more than 25% of people click the first search result they see. SEO helps you rank higher in search results and garner more visibility online, making potential customers more likely to click over to your site and convert.

Even if users don't visit and convert now, simply populating in the rankings will make users more familiar with your brand. In the future, they're more likely to remember your brand name the next time they're interested in associated products and services, and they may search for you directly.

Web Traffic

To put it simply—if potential customers can't find your website, you miss out on sales opportunities. SEO increases your organic search engine traffic, in turn increasing the number of visitors your website sees each day. This directly correlates to an increase in

sales—because the more relevant people see your site, the more chances you have to sell to them.

Trustworthy

The better optimized your site is, the higher you'll appear on search engines like Google and Bing. While ranking higher on Google is appealing to all brands because of increased visibility, a secondary benefit is the trust you gain with potential customers. Users tend to defer to the recommendations that a search engine generates, so having a higher position for the keywords a user is searching for will solidify your product or service as trustworthy in the user's mind.

User Experience

A well-optimized website clearly communicates what product or service is being offered, shows how to obtain it, and answers any questions surrounding it. User experience is a major ranking factor for Google. This means that by catering the site to appeal to search engines like Google and Bing, you're also catering it to the user's experience. This means both search engines and users are able to easily get the information they need. On the other hand, if a user struggles to navigate your site, chances are that search engines will as well.

Growth

There's no doubt about it—SEO carries a lot of importance for the growth of your brand. As we mentioned above, the higher you rank on a search engine for a variety of high-volume keywords, the more organic (aka non-paid) web traffic your site will receive. It's as simple as that.

A website that is well optimized is more likely to gain more customers and make more sales due to increased

lead generation. People are also more likely to share your brand across other social platforms like Facebook or Instagram once they've found your website through a search engine.

6. What is Accessibility?

This article starts the module off with a good look at what accessibility is — this overview includes what groups of people we need to consider and why, what tools different people use to interact with the web, and how we can make accessibility part of our web development workflow.

Accessibility is the practice of making your websites usable by as many people as possible. We traditionally think of this as being about people with disabilities, but the practice of making sites accessible also benefits other groups such as those using mobile devices, or those with slow network connections.

You might also think of accessibility as treating everyone the same, and giving them equal opportunities, no matter what their ability or circumstances. Just as it is wrong to exclude someone from a physical building because they are in a wheelchair (modern public buildings generally have wheelchair ramps or elevators), it is also not right to exclude someone from a website because they have a visual impairment. We are all different, but we are all human, and therefore have the same human rights.

Accessibility is the right thing to do. Providing accessible sites is part of the law in some countries, which can open up some significant markets that otherwise would not be able to use your services or buy your products.

Building accessible sites benefits everyone:

- Semantic HTML, which improves accessibility, also improves SEO, making your site more findable.
- Caring about accessibility demonstrates good ethics and morals, which improves your public image.
- Other good practices that improve accessibility also make your site more usable by other groups, such as mobile phone users or those on low network speed. In fact, everyone can benefit from many such improvements.
- Did we mention it is also the law in some places?

7. What is Markup Language?

A markup language is a set of rules that defines how the layout and presentation of text and images should appear in a digital document. It allows structuring documents, adding formatting, and specifying how different elements should be displayed (or “rendered”) on webpages.

This structuring helps search engines like Google understand the information on websites better. If search engines know more about what a page is about, they are more likely to show it to people who are looking for its content. Which, in result, can bring more people to websites with the right markup.

An example of a markup language—and the one most people know—is HTML. And it looks like this:

```
<b>Example of a Markup Language</b>
```

For now, note that markup languages are different from programming languages.Programming languages are used to create functional and dynamic web applications.

Markup languages focus primarily on the presentation and structure of content. They are static and don’t use logic or algorithms.

To give you an even better idea of what markup languages are, let’s look closer at two categories of markup languages: semantic and presentational markup.

8. What is HTML?

- HTML stands for Hyper Text Markup Language
- HTML is the standard markup language for creating Web pages
- HTML describes the structure of a Web page
- HTML consists of a series of elements
- HTML elements tell the browser how to display the content
- HTML elements label pieces of content such as "this is a heading", "this is a paragraph", "this is a link", etc.

```
<!DOCTYPE html>
<html>
<head>
<title>Page Title</title>
</head>
<body>

<h1>My First Heading</h1>
```

```
<p>My first paragraph.</p>

</body>
</html>
```

9. What is browser engine?

A web browser is a software application that lets you explore the internet. It retrieves and displays web pages, images, videos, and other content from web servers. Each piece of content has a unique address called a URL (Uniform Resource Locator), which tells the browser where to find it.

While we often focus on the browser itself (Chrome, Firefox, Edge, etc.), each choice also determines the underlying browser engine and rendering engine. These core components work together to create the web experience we see. Though sometimes used interchangeably, they have distinct roles. Here, In this article, we are going to study the browser engine and how it works under the hood.

Think of a browser engine as the heart of your web browser. It is the essential software that acts as a bridge between the web page's code (HTML, CSS, JavaScript) and the visual experience you see on your screen. Here's what it does:

- **Starts the Process:** The browser engine handles the initial request to load a web page and coordinates the resources needed.
- **Navigation:** It manages your browsing actions – things like going back, forward, or reloading.
- **Error Handling:** The browser engine gracefully displays error messages if something goes wrong.
- **Visual Layout:** It works behind the scenes, using HTML and CSS to calculate the precise position and appearance of every element on the page.

10. What is rendering engine? share the available rendering engine?

Understanding the rendering engine before understanding the browser engine is essential. A browser's rendering engine interprets the visual elements of a web page and displays them correctly, following the developer's intentions. It inputs HTML and CSS files and presents them accurately on the screen. Rendering engines are often called layout engines because they functionally read CSS files.

Blink: Google's Blink engine is part of the Chromium project and is used in all Chromium-based browsers, including Microsoft Edge and non-Chromium browsers like Opera. Blink was created to address WebKit's shortcomings in supporting a multi-process browser architecture.

WebKit: The rendering engine developed by Apple and used in their devices, including iPhones, is WebKit. It originated in 1998 and has been the primary foundation for Blink and other such projects. This is why many WebKit contributors, including Sony, used it on their PlayStation devices.

Gecko: Gecko is an open-source rendering engine developed by Mozilla that powers the Firefox browser. It is known for its adherence to web standards and strict compatibility with various web technologies. Gecko is a crucial component of Firefox's rendering capabilities. It is also used by multiple projects, including those not affiliated with Firefox, such as Android, macOS, and Windows.

11. What is JavaScript Engine? share the available JS engine? Purpose of JS Engine?

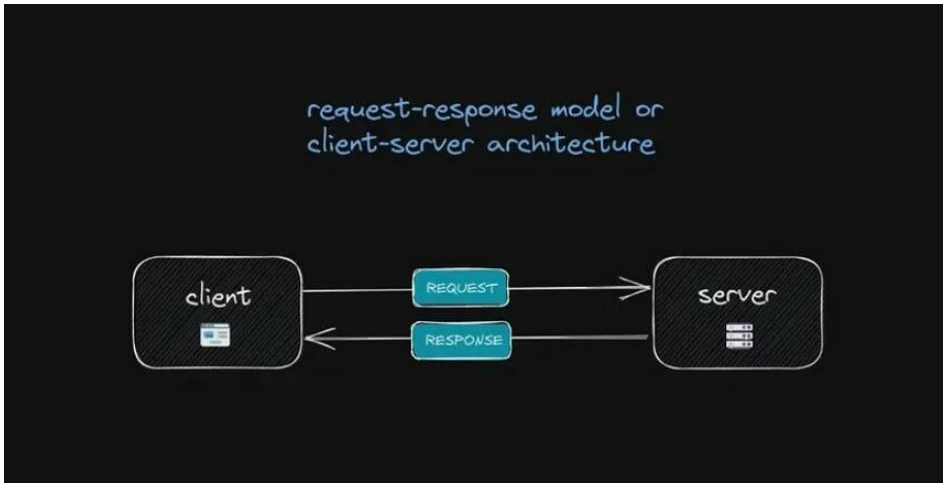
JavaScript is a scripting language and is not directly understood by computer but the browsers have inbuilt JavaScript engine which help them to understand and interpret JavaScript codes. These engines help to convert our JavaScript program into computer-understandable language. A JavaScript engine is a computer program that executes JavaScript code and converts it into computer understandable language.

1. **V8:** V8 is a JavaScript engine developed by the Chromium Project for Google Chrome and Chromium web browsers. It is a JavaScript engine that can run standalone, or be embedded into any C++ application. Using its own parser, it generates an abstract syntax tree.
2. **Spider Monkey:** SpiderMonkey is the first JavaScript engine, written by Brendan Eich at Netscape Communications, later released as open-source and currently maintained by the Mozilla Foundation. It is still used in the Firefox web browser.

A JavaScript engine is a software component that executes JavaScript code. The first JavaScript engines were mere interpreters, but all relevant modern engines use just-in-time compilation for improved performance. JavaScript engines are typically developed by web browser vendors, and every major browser has one.

12. How website works?

The process is quite simple. When we enter a URL into our browser, the browser sends a request for the necessary data to the server. The server then responds by sending back the code and data for the website. The browser interprets the codes and displays the website for us to view. This process is known as the request-response model or client-server architecture.



**Clients **are those devices connected to the internet, like our phones or computers connected to the mobile network or wi-fi. The client is where all the user interaction happens. In the web context, a client is typically a web-accessing software like a browser, such as Chrome, Firefox, or Safari, that requests web pages and other resources from a server. The client (web browser) receives the code for the website and then renders it for the user to view.

Though we access the website from the browser, we can treat the whole device as a client of the client-server architecture.

On the other end of the spectrum are **Servers**, specialized computers designed to store and manage data, websites, and web applications. These servers are called such because they serve up code or data in response to requests made by clients. A server waits for requests to come in from clients, processes the request, and then sends back the requested information. There are various types of servers, such as web servers, file servers, and database servers, each with its specific

function. In this article, we will primarily focus on web servers.

13. What is Data Structure?

Data structures serve as the basis for abstract data types (ADT). The ADT defines the logical form of the data type. The data structure implements the physical form of the data type.[5]

Different types of data structures are suited to different kinds of applications, and some are highly specialized to specific tasks. For example, relational databases commonly use B-tree indexes for data retrieval,[6] while compiler implementations usually use hash tables to look up identifiers.[7]

Data structures provide a means to manage large amounts of data efficiently for uses such as large databases and internet indexing services. Usually, efficient data structures are key to designing efficient algorithms. Some formal design methods and programming languages emphasize data structures, rather than algorithms, as the key organizing factor in software design. Data structures can be used to organize the storage and retrieval of information stored in both main memory and secondary memory.[8]

14. Explain Tree Data Structure?

Tree data structure is a specialized data structure to store data in hierarchical manner. It is used to organize and store data in the computer to be used more effectively. It consists of

a central node, structural nodes, and sub-nodes, which are connected via edges. We can also say that tree data structure has roots, branches, and leaves connected.

Tree data structure is a hierarchical structure that is used to represent and organize data in a way that is easy to navigate and search. It is a collection of nodes that are connected by edges and has a hierarchical relationship between the nodes. The topmost node of the tree is called the root, and the nodes below it are called the child nodes. Each node can have multiple child nodes, and these child nodes can also have their own child nodes, forming a recursive structure.
The data in a tree are not stored in a sequential manner i.e., they are not stored linearly. Instead, they are arranged on multiple levels or we can say it is a hierarchical structure. For this reason, the tree is considered to be a non-linear data structure.

15. What is user agent? share the list and its purpose?

As we browse the web, we've come to expect a seamlessly smooth experience from the websites we visit. And as website owners, we strive to improve the performance and engagement of our websites by understanding our visitors. We need to understand their different characteristics, from their location to the browser, operating system, and hardware of the device they are using to access our site.

This information supports a multitude of use cases, including website analytics and optimization, security features if your user logs in to your site with a new device, or content adaption tailored specifically to the device used (such as a mobile phone). These use cases all require knowledge of your user's device, browser, or operating system.

One way to gather this information is via the User-Agent – a type of HTTP request header. For nearly thirty years, the User-Agent has been a well-established part of the web experience. However, a new HTTP header has been created by Google to replace the User-Agent: User-Agent Client Hints (UA-CH).

We've got more information on User-Agent Client Hints later in this blog. In the meantime, let's dissect the User-Agent for more detail.

- Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/123.0.0.0 Safari/537.36
- Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:124.0) Gecko/20100101 Firefox/124.0
- Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/123.0.0.0 Safari/537.36 Edg/123.0.2420.81
- Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/123.0.0.0 Safari/537.36 OPR/109.0.0.0
- Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/123.0.0.0 Safari/537.36
- Mozilla/5.0 (Macintosh; Intel Mac OS X 14.4; rv:124.0) Gecko/20100101 Firefox/124.0
- Mozilla/5.0 (Macintosh; Intel Mac OS X 14_4_1) AppleWebKit/605.1.15 (KHTML, like Gecko) Version/17.4.1 Safari/605.1.15
- Mozilla/5.0 (Macintosh; Intel Mac OS X 14_4_1) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/123.0.0.0 Safari/537.36 OPR/109.0.0.0
- Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/123.0.0.0 Safari/537.36
- Mozilla/5.0 (X11; Linux i686; rv:124.0) Gecko/20100101 Firefox/124.0

1. Rotate User Agents

Rotating scraping user agents refers to changing them while making web requests. This lets you access more data and increases your scraper's efficiency. This method can help protect your IP address from getting blocked and blacklisted.

2. Keep Random Intervals Between Requests

Maintain random intervals between requests to prevent your spider from getting detected and blocked.

You might be interested in reading our guide on how to bypass rate limits while web scraping.

3. Use Up-to-date User Agents

To keep your scraping experience smooth and uninterrupted, ensure that your spider's user agents are regularly updated. Outdated ones can get your IP blocked!

16. What is Hypertest?

Hypertext is text displayed on a computer display or other electronic devices with references (hyperlinks) to other text that the reader can immediately access.[1] Hypertext documents are interconnected by hyperlinks, which are typically activated by a mouse click, keypress set, or screen touch. Apart from text, the term "hypertext" is also sometimes used to describe tables, images, and other presentational content formats with integrated hyperlinks. Hypertext is one of the key underlying concepts of the World Wide Web,[2] where Web pages are often written in the Hypertext Markup Language (HTML). As implemented on the Web, hypertext enables the easy-to-use publication of information over the Internet.

Hypertext documents can either be static (prepared and stored in advance) or dynamic (continually changing in response to user input, such as dynamic web pages). Static hypertext can be used to cross-reference collections of data in documents, software applications, or books on CDs. A well-constructed system can also incorporate other user-interface conventions, such as menus and command lines. Links used in a hypertext document usually replace the current piece of hypertext with the destination document. A lesser known feature is StretchText, which expands or contracts the content in place, thereby giving more control to the reader in determining the level of detail of the displayed document. Some implementations support transclusion, where text or other content is included by reference and automatically rendered in place.

Hypertext can be used to support very complex and dynamic systems of linking and cross-referencing. The most famous implementation of hypertext is the World Wide Web, written in the final months of 1990 and released on the Internet in 1991.

17. What is HTML Tags?

HTML tags are like keywords which defines that how web browser will format and display the content. With the help of tags, a web browser can distinguish between an HTML content and a simple content. HTML tags contain three main parts: opening tag, content and closing tag. But some HTML tags are unclosed tags.

When a web browser reads an HTML document, browser reads it from top to bottom and left to right. HTML tags are used to create HTML documents and render their properties. Each HTML tags have different properties.

An HTML file must have some essential tags so that web browser can differentiate between a simple text and HTML text. You can use as many tags you want as per your code requirement.

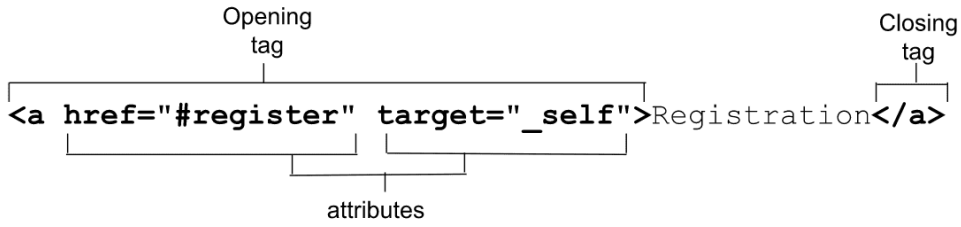
- All HTML tags must enclosed within < > these brackets.
- Every tag in HTML perform different tasks.
- If you have used an open tag <tag>, then you must use a close tag </tag> (except some tags)

Syntax

<tag> content </tag>

18. What is HTML Attributes?

Attributes are what make HTML so powerful. Attributes are space-separated names and name/value pairs appearing in the opening tag, providing information about and functionality for the element.



Attributes define the behavior, linkages, and functionality of elements. Some attributes are global, meaning they can appear within any element's opening tag. Other attributes apply to several elements but not all, while other attributes are element-specific, relevant only to a single element. In HTML, all attributes except boolean, and to some extent enumerated attributes, require a value.

If an attribute value includes a space or special characters, the value must be quoted. For this reason, and for improved legibility, quoting is always recommended.

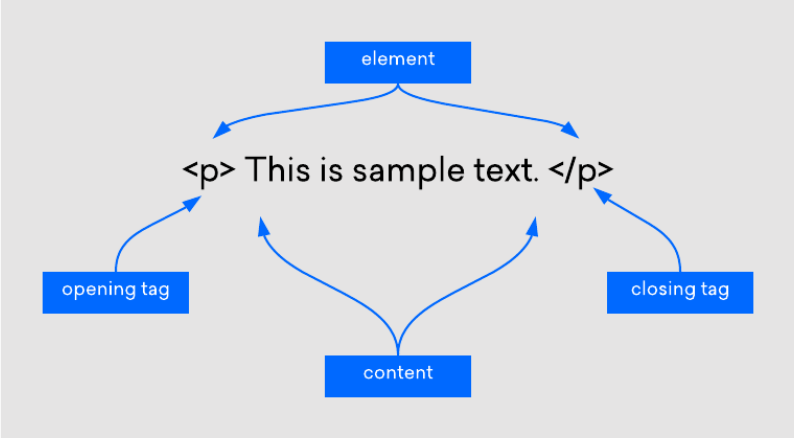
While HTML is not case-sensitive, some attribute values are. Values that are part of the HTML specification are case-insensitive. Strings values that are defined, such as class and id names, are case-sensitive. If an attribute value is case-sensitive in HTML, it is case-sensitive when used as part of an attribute selector in CSS and in JavaScript; otherwise, it's not.

<!-- the type attribute is case insensitive: these are equivalent -->

19. What is HTML Elements?

An HTML element is a component of an HTML document that tells a web browser how to structure and interpret a part of the HTML document. HTML elements can contain formatting instructions, semantic meaning, and content.

For example, HTML elements are used to denote document parts such as headers, paragraphs, and footers and to embed content such such as hyperlinks, text, and images. Although HTML can be used to provide formatting instructions, HTML standards strongly encourage using CSS for this purpose instead.
An HTML element is often — but not always — created by opening and closing HTML tags, which wrap around a piece of content. Below is an illustration that labels each of the parts of an HTML element:



20. How do convert elements to tree?

Converting elements into a tree typically involves constructing a data structure where each element becomes a node in the tree. The exact approach depends on the specifics of your elements and the desired structure of the tree (e.g., binary tree, n-ary tree).

Here’s a general approach to convert elements into a tree:

1. **Define the Tree Structure**: Decide on the type of tree you want (binary tree, n-ary tree, etc.), and understand the relationships between nodes (parent-child relationships).
2. **Choose a Representation**: You can represent a tree using nodes and pointers (in languages like Python, this might be implemented using classes and references) or using an adjacency list or matrix (in more abstract or mathematical contexts).
3. **Construct the Tree**:
    - Start with the root node. This will be your starting point.
    - For each element that you want to add to the tree, decide where it fits based on the tree's rules (e.g., in a binary search tree, smaller elements go to the left, larger to the right; in an n-ary tree, each node can have multiple children).
4. **Insert Elements**:
    - Begin with the root node.
    - Traverse the tree according to the rules of the tree structure until you find the appropriate place for the new element.
    - Insert the element as a new node at that position.
5. **Repeat for Each Element**: Continue inserting elements one by one until all elements have been added to the tree.

Example: Binary Search Tree (BST)

21. What is DOCTYPE?

In web development, DOCTYPE stands for "Document Type Declaration". It is an instruction to web browsers and validators about the version of HTML or XHTML that the web page is written in. The DOCTYPE declaration is not a tag; rather, it is a directive that must appear at the very beginning of an HTML or XHTML document, before the <html> tag.

Purpose of DOCTYPE

1. **Browser Rendering Mode**: DOCTYPE helps browsers to determine how to render a web page by specifying whether to use standards mode, quirks mode, or almost standards mode.
2. **Validation**: It allows validators to check the syntax and structure of the document against the specified document type.

Functionality of DOCTYPE

- Standards Mode: When a correct DOCTYPE declaration is included, browsers will render the page according to the specified HTML or XHTML standard, ensuring consistency across different browsers.
- Quirks Mode: If no DOCTYPE declaration is specified or an incorrect one is used, browsers may fall back to quirks mode, which emulates older, non-standard behavior from earlier versions of browsers. This can lead to inconsistent rendering across different browsers.

22. What are the ways we can save html file?

Saving an HTML file typically involves creating or editing the file using a text editor or an Integrated Development Environment (IDE), and then saving it to your local file system. Here are the common ways to save an HTML file:

1. Text Editor:
    - Notepad (Windows): Open Notepad, write or paste your HTML code, then go to File > Save As. Choose the location and filename, ensuring to select "All

Files" in the "Save as type" dropdown and append .html to the filename (e.g., index.html).
    - TextEdit (Mac): Similar process to Notepad. Write or paste your HTML code, then go to File > Save As. Choose the location and filename, ensuring to select "Plain Text" format and append .html to the filename.
    - Sublime Text, Visual Studio Code, Atom, etc.: These text editors typically have dedicated support for HTML syntax highlighting and often provide shortcuts or plugins to streamline saving HTML files.
2. Integrated Development Environment (IDE):
    - IDEs like Visual Studio, Eclipse, IntelliJ IDEA, and others are commonly used for web development. They provide features like syntax highlighting, auto-completion, and integrated file management.
    - Saving an HTML file in an IDE involves creating or editing a file within the project structure and saving it with an .html extension.
3. Using Command Line:
    - If you prefer using command-line interfaces (CLI):
        - Windows Command Prompt or PowerShell: Use text editors like notepad, edit, or echo with output redirection to create and save HTML files.
        - Unix/Linux Terminal: Use commands like nano, vim, emacs, or simply redirect output from echo or other commands to create and save HTML files.
4. Through Web Browsers:
    - You can also create and save HTML files directly through web browsers:
        - Chrome, Firefox, Edge, Safari: Open the browser, type or paste your HTML code into the address bar prefixed with data:text/html, (e.g., data:text/html,<html><body><h1>Hello World!</h1></body></html>), then right-click

and choose Save Page As to save it as an HTML file.

23. What is charset? why we need to use this?

In the context of web development, charset stands for character set. It refers to a specific encoding scheme that determines how characters are represented and stored in a computer's memory or on external media such as files. Here’s why charset is important and why we need to use it:

Importance of charset:

1. **Character Encoding**: Different character sets define how characters (letters, numbers, symbols) are encoded as binary data, which computers can understand and process. Common character encodings include UTF-8, ISO-8859-1, UTF-16, etc.
2. **Display and Interpretation**: The charset declaration specifies to the browser how the bytes of the HTML, CSS, or JavaScript files should be interpreted and displayed as text. Without a specified charset, the browser may not correctly interpret special characters or non-English text, leading to garbled or incorrectly displayed content.
3. **Internationalization (i18n)**: Websites and applications often need to support content in multiple languages and scripts. Using a proper charset ensures that text in various languages (e.g., English, Chinese, Arabic) is displayed correctly across different browsers and devices.
4. **Compatibility**: Modern browsers generally default to UTF-8 encoding if no charset is specified. However, specifying charset explicitly ensures consistent behavior across all browsers and prevents potential issues with older browsers or unusual configurations.

How to Specify charset:

- **HTML**: Specify the character set within the <meta> tag in the <head> section of your HTML document:
- This tells the browser to use UTF-8 encoding to interpret the characters in the HTML file.
- **HTTP Headers**: For server responses, you can specify the character set using HTTP headers. For example, in PHP, you can set the Content-Type header:
- This informs the browser that the server response is HTML content encoded in UTF-8.

24. What is meta data? what is the purpose of it?

1. **Definition**: Metadata in HTML is typically provided using <meta> tags within the <head> section of an HTML document. These tags do not display content on the web page itself but instead provide information such as character encoding, authorship, viewport settings, and more.
2. **Purpose**:
    - **Character Encoding**: <meta charset="UTF-8"> specifies the character encoding used in the HTML document. It ensures that special characters and non-English text are correctly interpreted and displayed by web browsers.
    - **Viewport Settings**: <meta name="viewport" content="width=device-width, initial-scale=1.0"> is used to control the layout and scaling of the web page on different devices (especially mobile devices). It ensures that the page renders correctly across various screen sizes.
    - **Page Description**: <meta name="description" content="Description of the page"> provides a concise summary or description of the content of the

web page. This description may be used by search engines as a snippet in search results.
    - **Keywords**: <meta name="keywords" content="keyword1, keyword2, ..."> specifies keywords relevant to the content of the page. While not heavily used by search engines anymore, it can still provide context about the page's content.
    - **Author**: <meta name="author" content="Author Name"> indicates the author or creator of the web page. It can be helpful for identifying the responsible entity for the content.
    - **Viewport**: <meta name="viewport" content="width=device-width, initial-scale=1.0"> adjusts the layout viewport to the width of the device, enhancing mobile responsiveness.
3. **Implementation**: <meta> tags are self-contained within the <head> section of an HTML document. They do not have closing tags and typically use attributes (name and content, or charset as an attribute) to convey specific information.

Importance of Metadata:

- **SEO**: Metadata like description and keywords can influence how search engines index and display the page in search results.
- **Accessibility**: Metadata such as viewport settings ensures that web pages are accessible and readable across different devices and screen sizes.
- **Interoperability**: Properly defined metadata enhances the interoperability of web content by providing standardized information that browsers and search engines can use uniformly.

25. Explain Web Application Architecture?

Web application architecture refers to the structure and organization of components and interactions within a web application. It defines how various software components (like servers, databases, client-side interfaces) work together to handle requests, process data, and deliver responses to users. Here's a comprehensive explanation of web application architecture:

Components of Web Application Architecture:

1. **Client-Side Components**:
    - **User Interface (UI)**: This includes HTML, CSS, and JavaScript that run in the user's browser. It handles the presentation layer and user interaction.
    - **Client-Side Frameworks**: Libraries like React, Angular, or Vue.js that facilitate building interactive user interfaces.
2. **Server-Side Components**:
    - **Web Server**: Handles incoming HTTP requests from clients (browsers), retrieves data from databases or other services, and sends back responses.
    - **Application Server**: Executes the logic of the application, such as processing business logic, interacting with databases, and generating dynamic content.
    - **Middleware**: Software components that bridge the gap between the client-side and server-side systems, like API gateways, authentication servers, and load balancers.
3. **Database**: Stores and manages the application's data. Common databases include SQL (e.g., MySQL, PostgreSQL) and NoSQL (e.g., MongoDB, Redis) depending on the requirements of the application.
4. **Networking**: Protocols and technologies (e.g., HTTP, HTTPS, RESTful APIs) that enable communication between clients, servers, and other services.

5. **Security**: Measures such as encryption, authentication, and authorization mechanisms to protect data and ensure secure communication between components.